

## Microsoft\_CertifyMe\_70-536\_v2011-01-12\_281q\_By-Gaahl

Number: 70-536

Passing Score: 800

Time Limit: 1200 min

File Version: 2011-01-12

***Microsoft : 70-536***

***Version : 2011-01-12***

***Question : 281***

***Good Luks  
Everiwun!***

***By-Gaahl***

## Exam B

### QUESTION 1

You are testing a newly developed method named PersistToDB. This method accepts a parameter of type EventLogEntry.

This method does not return a value. You need to create a code segment that helps you to test the method.

The code segment must read entries from the application log of local computers and then pass the entries on to the PersistToDB method.

The code block must pass only events of type Error or Warning from the source MySource to the PersistToDB method.

Which code segment should you use?

- A. 

```
EventLog myLog = new EventLog("Application", ".");
foreach (EventLogEntry entry in myLog.Entries){
    if (entry.Source == "MySource"){
        PersistToDB(entry);
    }
}
```
- B. 

```
EventLog myLog = new EventLog("Application", ".");
myLog.Source = "MySource";
foreach (EventLogEntry entry in myLog.Entries){
    if (entry.EntryType == (EventLogEntryType.Error & EventLogEntryType.Warning)){
        PersistToDB(entry);
    }
}
```
- C. 

```
EventLog myLog = new EventLog("Application", ".");
foreach (EventLogEntry entry in myLog.Entries){
    if (entry.Source == "MySource"){
        if (entry.EntryType == EventLogEntryType.Error || entry.EntryType == EventLogEntryType.Warning){
            PersistToDB(entry);
        }
    }
}
```
- D. 

```
EventLog myLog = new EventLog("Application", ".");
myLog.Source = "MySource";
foreach (EventLogEntry entry in myLog.Entries){
    if (entry.EntryType == EventLogEntryType.Error || entry.EntryType == EventLogEntryType.Warning){
        PersistToDB(entry);
    }
}
```

**Answer:** C

**Section:** (none)

#### Explanation/Reference:

Explanation: It is necessary to create a new Application EventLog, iterate over all the EventLogEntries and call the PersistToDB method

if the entry is a warning or error and the source is MySource.

A will PersistToDB irrespective of the type of log entry. The question explicitly states only warnings and errors should be persisted.

B features an incorrect test for warnings and errors.

D&B do not ensure that only MySource entries are persisted. Instead they overwrite the source.

· Achtung Source muss manuell abgefragt werden

## QUESTION 2

You need to write a code segment that performs the following tasks: ? Retrieves the name of each paused service. ?

Passes the name to the Add method of Collection1.

Which code segment should you use?

- A. 

```
ManagementObjectSearcher searcher = new ManagementObjectSearcher( "Select * from Win32_service where State = 'Paused'");
foreach (ManagementObject svc in searcher.Get()) {
    Collection1.Add(svc["DisplayName"]);
}
```
- B. 

```
ManagementObjectSearcher searcher = new ManagementObjectSearcher( "Select * from Win32_service", "State = 'Paused'");
foreach (ManagementObject svc in searcher.Get()) {
    Collection1.Add(svc["DisplayName"]);
}
```
- C. 

```
ManagementObjectSearcher searcher = new ManagementObjectSearcher( "Select * from Win32_service");
foreach (ManagementObject svc in searcher.Get()) {
    if ((string) svc["State"] == "Paused") {
        Collection1.Add(svc["DisplayName"]);
    }
}
```
- D. 

```
ManagementObjectSearcher searcher = new ManagementObjectSearcher();
searcher.Scope = new ManagementScope(Win32_service");
foreach (ManagementObject svc in searcher.Get()) {
    if ((string)svc["State"] == "Paused") {
        Collection1.Add(svc["DisplayName"]);
    }
}
```

**Answer:** A

**Section:** (none)

### Explanation/Reference:

Explanation: Use the ManagementObjectSearcher to search for all services with a paused state. Iterate over the returned collection and add the display name to Collection1.

B The constructor is invoked incorrectly.

C & D the query is incorrect. The searcher does not restrict to paused services.

· C würde gehen aber mit mehr Aufwand als A !!!

## QUESTION 3

You are using the Microsoft Visual Studio 2005 IDE to examine the output of a method that returns a string.

You assign the output of the method to a string variable named fName. You need to write a code segment that prints the following on a single line The message:

"Test Failed: "

The value of fName if the value of fName does not equal "Certkiller" You also need to ensure that the code segment simultaneously facilitates uninterrupted execution of the application.

Which code segment should you use?

- A. 

```
Debug.Assert(fName == "CertKiller", "Test Failed: ", fName);
```
- B. 

```
Debug.WriteLineIf(fName != "CertKiller", fName, "Test Failed: ");
```

- C. 

```
if (fName != "Certkiller") {  
    Debug.print("Test Failed: ");  
    Debug.print(fName);  
}
```
- D. 

```
if (fName != "Certkiller") {  
    Debug.WriteLine("Test Failed: ");  
    Debug.WriteLine(fName);  
}
```

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

Explanation: `Debug.WriteLineIf()` will conditionally write the "Test Failed", it will not interrupt execution of the application.

A an Assert will stop execution of the application in debug mode if the condition is not met.

C & D could be used but they execute in the release configurations

· ACHTUNG: uninterrupted execution !!!

**QUESTION 4**

You are testing a method that examines a running process.

This method returns an `ArrayList` containing the name and full path of all modules that are loaded by the process.

You need to list the modules loaded by a process named `C:\TestApps\Process1.exe`.

Which code segment should you use?

- A. 

```
ArrayList ar = new ArrayList();  
Process[] procs;  
ProcessModuleCollection module;  
procs = Process.GetProcesses(@"Process1");  
if (procs.Length > 0) {  
    module = procs[0].Modules;  
    foreach (ProcessModule mod in modules) {  
        ar.Add(mod.ModuleName);  
    }  
}
```
- B. 

```
ArrayList ar = new ArrayList();  
Process[] procs;  
ProcessModuleCollection module;  
procs = Process.GetProcesses("C:\TestApps\Process1.exe");  
if (procs.Length > 0) {  
    module = procs[0].Modules;  
    foreach (ProcessModule mod in modules) {  
        ar.Add(mod.ModuleName);  
    }  
}
```

- C. 

```
ArrayList ar = new ArrayList();
Process[] procs;
ProcessModuleCollection module;
procs = Process.GetProcessesByName(@"Process1");
if (procs.Length > 0) {
    module = procs[0].Modules;
    foreach (ProcessModule mod in modules) {
        ar.Add(mod.FileName);
    }
}
```
- D. 

```
ArrayList ar = new ArrayList();
Process[] procs;
ProcessModuleCollection module;
procs = Process.GetProcessesByName(@"C:\TestApps\Process1.exe");
if (procs.Length > 0) {
    module = procs[0].Modules;
    foreach (ProcessModule mod in modules) {
        ar.Add(mod.FileName);
    }
}
```

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

Explanation: Process.GetProcessesByName() should be used to return all the processes that match a process name.

The modules collection exposes all the modules loaded by the process and can be added to an ArrayList.

A & B GetProcesses() accepts a computer name for retrieving the processes on a remote computer. GetProcessesByName() should be used to return processes by their name. D the path of the process is not part of the process name.

·Process.GetProcessesByName("nurName");

**QUESTION 5**

You have been asked to profile a business application that can be accessible using the Event Log API.

You have started by adding the following code to create a custom event log:

```
if (EventLog.SourceExists ("Application1"))
EventLog.DeleteEventSource ("Application1");
//Create new event log
EventLog.CreateEventSource ("Application1", "Profile");
```

You need to write an event to the Application1 event log.

What code must you use?

- A. 

```
EventLog log = new EventLog ();
log.Source = "Application1";
log.Log = "Profile";
log.WriteEvent("Writing to event log.");
```
- B. 

```
EventLog log = new EventLog ();
log.Source = "Profile";
log.Log = "Application1";
log.WriteEvent("Writing to event log.");
```

- C. `EventLog log = new EventLog ();`  
`log.Source = "Application1";`  
`log.Log = "Profile";`  
`log.WriteEntry("Writing to event log.");`
- D. `EventLog log = new EventLog ();`  
`log.Source = "Profile";`  
`log.Log = "Application1";`  
`log.WriteEntry("Writing to event log.");`

**Answer:** C

**Section:** (none)

#### **Explanation/Reference:**

This code instantiates an EventLog object, sets the Source and Log properties, and invokes the WriteEntry method

to output the message to the event log.

The EventLog object allows you to create, delete, read from, or write to Windows event logs.

In this scenario, you use the CreateEventSource method to create a custom event log. When calling the CreateEventSource method, you pass the method two arguments. The first argument represents the source name for the event log, and the second represents the name of the event log.

Next, you want to write an entry to the Application1 event log. To write to an event log, you must first identify the event source and the name of the event log to which you want to write.

The Source property specifies the event source, and the Log property specifies the name of the event log.

There are two methods to write to the EventLog object:

WriteEntry and WriteEvent.

The WriteEntry method is an overloaded method used to write a text message to an event log.

The WriteEvent method is used to write localized resources and event instances to an event log.

Incorrect Answers:

A, B: The WriteEvent method is used to write localized resources and event instances to an event log.

D: You should not use the code fragments that specify a value of "Profile" for the Source property and a value "Application1" for the Log property because they would attempt to write the entry to an event log named Application1.

- Source und Log - Eigenschaft richtig belegen und log.WriteEntry(string);
- C. `EventLog log = new EventLog ();`  
`log.Source = "Application1";`  
`log.Log = "Profile";`  
`log.WriteEntry("Writing to event log.");`

· A ist falsch wegen

· log.WriteEvent - benötigt eine EventInstanz !!!

#### **QUESTION 6**

The Certkiller .com network contains an application server named Certkiller -SR07.

You have been asked to profile a business application that can be accessible using the Event Log API.

You want to achieve this by creating a custom event log on Certkiller -SR07.

What should you do?

- A. Use the following code:  
`EventLog.CreateEventSource("Application1", "Profile", "Certkiller -SR07");`
- B. Use the following code:  
`EventLog.CreateEventSource("Application1", "Profile");`

- C. Use the following code:  
`EventSourceCreationData sourceData = new EventSourceCreationData("Application1", "Profile");  
sourceData.MachineName = "Certkiller -SR07";  
EventLog.CreateEventSource (sourceData);`
- D. Use the following code:  
`EventSourceCreationData sourceData = new EventSourceCreationData ("Application1", "Profile");  
EventLog.CreateEventSource(sourceData);`

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

Explanation: This code instantiates an EventSourceCreationData object, sets the MachineName property of the EventSourceCreationData object, and invokes the CreateEventSource method, passing the EventSourceCreationData object

as an argument, to create the custom event log.

The EventSourceCreationData object is used to configure a new event log source. You can then pass the EventSourceCreationData object to the CreateEventSource method of an EventLog

to register the event log source and corresponding event log so that you can write entries to it.

The EventSourceCreationData object's constructor accepts two String arguments: Source and Log.

The Source argument specifies the event source, and the Log argument specifies the name of the event log.

After creating an instance of EventSourceCreationData, you can set the EventSourceCreationData object's properties to further configure the event source.

The MachineName property represents the computer on which you want to create the event source.

In this scenario, you create a new EventSourceCreationData object named sourceData and then set its MachineName property to Certkiller -SR07. Then, you call the CreateEventSource method, passing the method sourceData. This creates an event log source on the computer named Certkiller -SR07.

Incorrect Answers:

A: The CreateEventSource method that takes three String arguments is obsolete in the .NET 2.0 Framework.

B, D: Both of these code fragments would create an event log on the local computer by default, but the scenario states that you should explicitly specify the machine name because the computer on which the code is running is unknown.

· Tipp: wo wird der Server angegeben (A und C)

· CreateEventSource ist veraltet !!! - daher ist A falsch · EventSourceCreationData soll verwendet werden daher C richtig.

· public EventSourceCreationData (string source, string logName) · Standardmässig der lokale Computer.

· Änderung über die Eigenschaft MachineName möglich.

**QUESTION 7**

You are required to retrieve and display the names of all processes that are currently running in memory. What should you do?

- A. Use the following code:  
`foreach (Process curProcess in Process.GetSystemProcesses())  
Console.WriteLine (curProcess.ProcessName);`
- B. Use the following code:  
`foreach (Process curProcess in Process.GetAllProcesses())  
Console.WriteLine (curProcess.ProcessName);`
- C. Use the following code:  
`foreach (Process curProcess in Process.GetProcesses())  
Console.WriteLine (curProcess.ProcessName);`

- D. Use the following code:
- ```
foreach (Process curProcess in Thread.GetProcesses())  
    Console.WriteLine (curProcess.ProcessName);
```

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

The GetProcesses method creates and returns an array of Process objects representing all currently running processes in memory.

Because this method invocation does not contain a computer name, the local machine is assumed.

The curProcess variable is reassigned to a Process object with each iteration. The ProcessName property of the Process object retrieves the system name of the process.

The WriteLine method of the Console class outputs the process name to the console. Because the WriteLine method outputs a line return, each process name will be on a separate line in the Console window.

Incorrect Answers:

A, B: The GetSystemProcesses and GetAllProcesses methods do not exist in the Process class.

D: There is no such method as the GetProcesses method in the Thread class.

· Process.GetProcesses !!!

**QUESTION 8**

Certkiller.com is currently using a performance counter named HitCounter.

You are required to increment each time a user accesses the UI classes within the application.

To do this, you start by creating performance counters using the following code:

GRAFIK - wird nicht für Antwort benötigt !!!

You have to adjust the HitCounter performance counter when a hit occurs.

What should you do?

- A. Use the following code:
- ```
PerformanceCounter hitCounter = new PerformanceCounter ("ApplicationPerformance", "HitCounter");  
hitCounter.ReadOnly = false;  
hitCounter.Increment ();
```
- B. Use the following code:
- ```
PerformanceCounter hitCounter = new PerformanceCounter ("ApplicationPerformance", "HitCounter");  
hitCounter.Increment ();
```
- C. Use the following code:
- ```
PerformanceCounter hitCounter = new PerformanceCounter ("ApplicationPerformance", "HitCounter");  
hitCounter ++;
```
- D. Use the following code:
- ```
PerformanceCounter hitCounter = new PerformanceCounter ("ApplicationPerformance", "HitCounter");  
hitCounter.ReadOnly = false;  
hitCounter ++;
```

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

Explanation: This code creates a new PerformanceCounter object that references the ApplicationPerformance category and the HitCounter performance counter. By default, a performance counter is read-only, so this code sets the ReadOnly property to False. Then, the Increment method is called to increase the counter by one. The PerformanceCounter class also provides an IncrementBy method, which increments the performance counter by the value specified in a provided argument.



Incorrect Answers:

B: You should not use the code fragments that fail to set the ReadOnly property to False because by default, all performance counter objects are read-only. C, D: You should not use the code fragments that use the ++ operator to increment the PerformanceCounter object because this will cause a compile-time error.

· PerformanceCounter wird standardmässig readonly erzeugt und hat eine Increment- bzw. Decrement-Methode.

· ReadOnly, Increment !!!

### QUESTION 9

You would like to create a custom performance counter for an application that you created recently.

To do this, you decide to create a custom category named ApplicationPerformance and a performance counter named HitNumber.

You need to ensure that this counter is shared across numerous applications.

What should you do?

A. Use the following code:

```
PerformanceCounterCategory.Create ("ApplicationPerformance", "ApplicationPerformanceHelp",  
PerformanceCounterCategoryType.MultipleInstance, "HitNumber", "HitNumberHelp");
```

B. Use the following code:

```
PerformanceCounterCategory.Create ("HitNumber", "HitNumberHelp",  
PerformanceCounterCategoryType.SingleInstance, "ApplicationPerformance",  
"ApplicationPerformanceHelp");
```

C. Use the following code:

```
PerformanceCounterCategory.Create ("HitNumber", "HitNumberHelp",  
PerformanceCounterCategoryType.MultipleInstance, "ApplicationPerformance",  
"ApplicationPerformanceHelp");
```

D. Use the following code:

```
PerformanceCounterCategory.Create ("ApplicationPerformance", "ApplicationPerformanceHelp",  
PerformanceCounterCategoryType.SingleInstance,  
"HitNumber", "HitNumberHelp");
```

**Answer:** D

**Section:** (none)

### Explanation/Reference:

Explanation: This code invokes the Create method of the PerformanceCounterCategory class. This method registers a custom category on the local computer. The first two arguments specify the name of the category and category help string, respectively.

The third argument specifies a PerformanceCounterCategoryType enumeration value indicating how many counters can be in use within the specified category.

The value PerformanceCounterCategoryType.SingleInstance indicates that only a single instance can exist for the entire category.

The last two arguments specify the name of the performance counter and the counter's help string, respectively.

Incorrect Answers:

A, C: The code fragments that specify the MultipleInstance value should not be used because the scenario requires you to share the same counter across multiple applications.

B: This code fragment incorrectly assigns the value HitNumber to the category name and ApplicationPerformance to the counter name.

· PerformanceCounterCategoryType.SingleInstance: Die Leistungsindikatorekategorie kann nur eine einzelne Instanz haben.

- PerformanceCounterCategoryType.MultipleInstance: Die Leistungsindikatorekategorie kann mehrere Instanzen haben.
- PerformanceCounterCategory.Create ("KategorieName", "KategorieHilfe", PerformanceCounterCategoryType.SingleInstance, "CounterName", "CounterHilfe");
- Tipp für Reihenfolge - was ist in der Hierarchie höher, das kommt zuerst (eine Kategorie hat einen Counter - nicht umgekehrt)

### QUESTION 10

You are required to launch the App.exe process, and specify sample.txt as the input file. App.exe will then use the contents of sample.txt to establish the environment's current settings. You need to ensure that the code you use meets these requirements. What should you do?

- Use the following code:  

```
Process myProcess = new Process ();
myProcess.StartInfo = new ProcessStartInfo("App.exe sample.txt");
myProcess.Start ();
```
- Use the following code:  

```
Process myProcess = new Process ();
myProcess.StartInfo = new ProcessStartInfo("App.exe");
myProcess.ProcessStartArgs ("sample.txt");
myProcess.Start();
```
- Use the following code:  

```
Process myProcess = new Process ();
myProcess.StartInfo = new ProcessStartInfo("App.exe");
myProcess.Start("sample.txt");
```
- Use the following code:  

```
Process myProcess = new Process ();
myProcess.StartInfo = new ProcessStartInfo("App.exe");
myProcess.StartInfo.FileName = "sample.txt";
myProcess.Start();
```

**Answer:** D

**Section:** (none)

### Explanation/Reference:

This code creates a new Process object, sets the application to run using the StartInfo property, and uses the FileName property to specify the file to use as input. The StartInfo property of a Process object stores startup information for the process. This includes the application to run and any command-line arguments to be used by the application.

The FileName property must be set, and can be set by either explicitly setting the FileName property or by passing the name of the file to the ProcessStartInfo constructor. The Start method launches the application process with the values specified in the StartInfo property.

In this scenario, the code declares a new Process object named myProcess and then sets the StartInfo property of this process.

The code passes the constructor of the ProcessStartInfo a single argument representing the name of the application, App.exe.

Next, the FileName property is explicitly set with an assignment statement to the value of "sample.txt".

Then, the code call the Start method to launch the application using the values specified in the customized startup settings.

Falsch:

A: You should not use the code that passes a single string containing the application and input file to the ProcessStartInfo constructor.

B: You should not use the code that invokes the ProcessStartArgs method because no such method exists in the Process class.

C: You should not use the code that invokes the Start method of the Process object specifying the input file because no such method signature exists.

- myProcess.ProcessStartInfo ("App.exe");
- myProcess.StartInfo.FileName = "sample.txt"

- ACHTUNG Antwort C:

- myProcess.Start ("sample.txt"); !!! nur die parameterlose Variante myProcess.Start() ist eine Instanzmethode !!!

- ALLE Aufrufe mit Parameter sind STATIC !!!

- z.B. Process.Start("sample.txt")

## QUESTION 11

You work as an application developer at Certkiller .com. Certkiller .com uses an application that calculates monthly payments based upon client input.

You are currently debugging this application using the Microsoft Visual Studio 2005 IDE.

The application contains the following code:

```
public double CalculateMonthlyPayment (Single rate, double principal) {  
//Implementation code  
}
```

You have discovered that unexpected results are being returned by the application. You would like to pause execution and display a message box containing an error message in the event of a negative or zero rate value is Passed to the CalculateMonthlyPayment method.

You need to ensure that this only occurs during debugging mode.

What should you do?

A. Add the following code to the beginning of the CalculateMonthlyPayment method:

```
Debug.Assert(rate > 0, "Rate Error", "Rate must be > zero");
```

B. Add the following code to the beginning of the CalculateMonthlyPayment method:

```
if (rate <= 0)  
    MessageBox.Show("Rate is" + rate, Error);
```

C. Add the following code to the beginning of the CalculateMonthlyPayment method:

```
if (rate <= 0)  
    Debug.WriteLine("Error Rate is" + rate);
```

D. Add the following code to the beginning of the CalculateMonthlyPayment method:

```
Debug.WriteLineIf(rate <= 0, "Error Rate is" & rate);
```

**Answer:** A

**Section:** (none)

### Explanation/Reference:

Explanation: This code makes the debugging assertion that the rate argument is greater than zero.

If it is not, then a message box will be display with the message Rate Error along with the detailed description "Rate must be > zero".

The Assert method of the debug class is an overloaded method that provides you the ability to test assumptions made in your programming logic.

The Assert method accepts three arguments, the first of which is required. This first argument represents a condition that is assumed to be true for your programming logic and will evaluate to a Boolean value.

The other two arguments represent optional string messages. When the Assert method is invoked with three arguments, the condition is evaluated. If the condition evaluates to true, then the program continues to execute. If the condition evaluates to false, the program execution is halted, and by default a modal dialog box is displayed.

This dialog box displays the first string on a single line, the second string on the second line, and then the location at which the assertion failed.

From this dialog box, you can invoke the debugger, continue execution, or exit the application.

Output generated using the Debug class is stripped out when creating a release version build of an application.

Incorrect Answers:

B: In this scenario, you want the dialog box to display only for debug builds of the application.

Using this code, the message box would display in a release build of the application. C, D: The output of these options will not display in a message box.

· Angabe: ...display a message box...

## QUESTION 12

You have recently created an application, and want to capture all debugging text messages generated by it. You would like these debugging messages to display on the command line. The application that you created contains the following code:

```
Debug.WriteLine ("Start the processing");
```

```
Console.WriteLine ("Generated by Console.WriteLine");
```

```
Debug.WriteLine ("End the processing");
```

You need to ensure that you are able to capture all debugging messages to the command line.

What should you do?

- A. Use the following code:  

```
Debug.Listeners.Add (new TextWriterTraceListener(Console.Out));  
Debug.AutoFlush = true;
```
- B. Use the following code:  

```
Debug.Listeners.Add (new StreamWriter(Console.Out));  
Debug.AutoFlush = true;
```
- C. Use the following code:  

```
Debug.Listeners.Add (new ConsoleTraceListener());  
Debug.AutoFlush = true;
```
- D. Use the following code:  

```
Debug.Listeners.Add (new DefaultTraceListener());  
Debug.AutoFlush = true;
```

**Answer:** C

**Section:** (none)

### Explanation/Reference:

Explanation: Listeners are used to capture trace and debug messages. Both the Trace and Debug objects share the same Listeners collection, which includes a DefaultTraceListener that will capture trace and debug messages in the Output window. You are able to override this behavior by using the Add method to add another listener to the Listeners collection.

The ConsoleTraceListener is used to route trace and debug messages to the console. The AutoFlush property should be set to true for immediate capture. This specifies that after each message is written, the buffer is flushed and the output is written to the listener.

Incorrect Answers:

A: You should use the code that instantiates a TextWriterTraceListener object that specifies the Console's output stream because this is less efficient than specifying a ConsoleTraceListener object.

B: You should use the code that instantiates a StreamWriter object that specifies the Console's output stream because the Listeners collection allows only Listener object streams.

D: You should use the code that instantiates a DefaultTraceListener object because this will capture the debugging messages

to the Output window in the Visual Studio .NET 2005 IDE. Also, each Listeners collection will contain a DefaultTraceListener by default.

### QUESTION 13

You have created an application to interface with an inventory storage system, and want to enable tracing in the application to track stock shortages and surpluses.

The application should not, however, trace inventory levels if no shortages or surpluses exist.

In addition, you do not want recompile every time to switch tracing capabilities on or off.

What should you do?

- A. Specify a command-line argument named InventoryTrace, and configure the application to read the argument.
- B. Specify a system-wide environmental variable named InventoryTrace, and configure the application to read the environmental variable.
- C. Specify a registry key named InventoryTrace, and configure the application to read the registry key setting.
- D. Specify a **BooleanSwitch** named InventoryTrace, and configure it in the application configuration file.

**Answer:** D

**Section:** (none)

#### Explanation/Reference:

Explanation: The BooleanSwitch class is used to toggle trace messages on and off. The application configuration is intended to manage any application-specific settings, including tracing.

The value of the Enabled property determines whether the BooleanSwitch is turned on or off. This value is set in the application configuration file. If the value is 0, then the BooleanSwitch object is turned off and the Enabled property returns false.

If the value is any other value, the BooleanSwitch object is turned on and the Enabled property returns true.

Incorrect Answers:

A: This option requires the application to run from the command-line.

B: This option requires EnvironmentPermission and is intended for more complex application settings.

C: This option requires RegistryPermission and is intended for system-wide settings, not for application-specific settings.

· D - da alle anderen Möglichkeiten Einschränkungen haben (Berechtigung bzw. Start eines Konsolenprogramms)

### QUESTION 14

Certkiller .com has a server named Certkiller -SR05 that has numerous processors installed.

You have been given the task of developing an application that displays certain clock speed statistics on all processors installed on Certkiller -SR05.

You need to ensure that this requirement is fully satisfied.

What should you do?

- A. Use the following code:

```
ManagementObjectSearcher processorSearcher = new ManagementObjectSearcher();
foreach (ManagementObject obj in processorSearcher.Get("SELECT * FROM Win32_Processor")){
    Console.WriteLine ("{0}", obj ["Name"]);
    Console.WriteLine ("{0} / {1}",
        obj ["CurrentClockSpeed"], Obj ["MaxClockSpeed"]);
}
```

- B. Use the following code:
- ```
ManagementObjectSearcher processorSearcher = new ManagementObjectSearcher( "SELECT * FROM Win32_Processor");
foreach (ManagementObject obj in processorSearcher.Get()){
    Console.WriteLine ("{0}", obj["Name"]);
    Console.WriteLine ("{0} / {1}",
        obj["CurrentClockSpeed"], obj["MaxClockSpeed"]);
}
```
- C. Use the following code:
- ```
ManagementObjectQuery processorQuery = new ManagementObjectQuery();
foreach (ManagementObject obj in processorQuery.Get( "SELECT * FROM Win32_Processor")){
    Console.WriteLine ("{0}", obj["Name"]);
    Console.WriteLine ("{0} / {1}",
        obj["CurrentClockSpeed"], obj["MaxClockSpeed"]);
}
```
- D. Use the following code:
- ```
ManagementObjectQuery processorQuery = new ManagementObjectQuery( "SELECT * FROM Win32_Processor");
foreach (ManagementObject obj in processorQuery.Get()){
    Console.WriteLine ("{0}", obj ["Name"]);
    Console.WriteLine ("{0} / {1}",
        obj["CurrentClockSpeed"], obj["MaxClockSpeed"]);
}
```

**Answer: B**

**Section: (none)**

#### **Explanation/Reference:**

This code retrieve all Win32\_Processor objects on the local machine, iterates through each Win32\_Processor object, and displays the Name, CurrentClockSpeed and MaxClockSpeed properties for each Win32\_Processor object. First, the ManagementObjectSearcher object is instantiated with the WQL query string on which to search. WQL is a subset of SQL, and it is specifically designed for WMI. The WQL string specified in this code returns any 32-bit processors available on the local machine.

The Get method returns a ManagementObjectCollection object. On each iteration, a ManagementObject object is assigned to the variable obj. The Console.WriteLine method display the Name, ClockSpeed, and MaxClockSpeed properties on the command-line using a string indexer.

Incorrect Answers:

A: You should not use the code that specifies the WQL query string as an argument of the Get method of the ManagementObjectSearcher object BECAUSE the Get method does not accept a string data type.

C D: You should not use the code that specifies the ManagementObjectQuery class BECAUSE there is no such class in the .NET 2.0 class library.

- new ManagementObjectSearcher( "SELECT \* FROM Win32\_Processor"); // WQL in Konstruktor
- foreach (ManagementObject obj in processorSearcher.Get()){ ...} // Get ohne Parameter!
- ManagementObjectQuery GIBT ES NICHT !!!

#### **QUESTION 15**

You have developed an application that simplifies hard drive management for Certkiller .com's administrators. The application that you created logs all modifications made to physical hard drives on an hourly basis. You now need to create an EventQuery object for this application. What should you do?

- A. Use the following code:  

```
EventQuery query = new EventQuery ();
query.QueryString = "SELECT InstanceModificationEvent FROM 'Win32_DiskDrive' WITHIN 3600";
```
- B. Use the following code:  

```
EventQuery query = new EventQuery ();
query.QueryString = "SELECT * FROM _InstanceModificationEvent WITHIN 3600 " + "WHERE TargetInstance ISA 'Win32_DiskDrive'";
```
- C. Use the following code:  

```
EventQuery query = new EventQuery ();
query.QueryString = "SELECT * FROM 'Win32_DiskDrive' WITHIN 3600" + "WHERE TargetInstance ISA _InstanceModificationEvent";
```
- D. Use the following code:  

```
EventQuery query = new EventQuery ();
query.QueryString = "SELECT * FROM _InstanceModificationEvent WITHIN 3600" + "WHERE Instance = 'Win32_DiskDrive'";
```

**Answer: B**

**Section: (none)**

**Explanation/Reference:**

Explanation: This code creates a new EventQuery object that queries all disk drive events every hour. The QueryString property contains a string that follows the syntax of WQL. WQL is a subset of SQL, and it is specifically designed for WMI. The SELECT clause indicates retrieval, where the asterisk (\*) represents all properties associated with the event.

The FROM clause indicates the event, which in this case is \_InstanceModificationEvent. The WITHIN clause specifies the polling interval in seconds. The WHERE clause specifies the condition that must be met. In this scenario, the TargetInstance field is compared to the value Win32\_DiskDrive. This limits the \_InstanceModificationEvent events to only those that modify hard drives.

Incorrect Answers:

A: InstanceModificationEvent is not a valid event property and Win32\_DiskDrive is not a valid event.

C: The FROM clause should specify the event InstanceModificationEvent and the ISA operator should use the Win32\_DiskDrive class.

D: Instance is not a valid field name and the equal sign (=) is not valid for class comparisons.

· WHERE

· ISA operator

· Use the ISA operator in the WHERE clause of a data query to request embedded objects in a class hierarchy.

· SELECT \* FROM ClassA WHERE EmbeddedObj ISA

'ClassZ'

**QUESTION 16**

You are writing a method to compress an array of bytes.

The array is passed to the method in a parameter named document.

You need to compress the incoming array of bytes and return the result as an array of bytes.

Which code segment should you use?

- A. 

```
MemoryStream strm = new MemoryStream(document);
DeflateStream deflate = new DeflateStream(strm, CompressionMode.Compress);
byte[] result = new byte[document.Length];
deflate.Write(result, 0, result.Length);
return result;
```



- B. 

```
MemoryStream strm = new MemoryStream(document);
DeflateStream deflate = new DeflateStream(strm, CompressionMode.Compress);
deflate.Write(document, 0, document.Length);
deflate.Close();
return strm.ToArray();
```
- C. 

```
MemoryStream strm = new MemoryStream();
DeflateStream deflate = new DeflateStream(strm, CompressionMode.Compress);
deflate.Write(document, 0, document.Length);
deflate.Close();
return strm.ToArray();
```
- D. 

```
MemoryStream inStream = new MemoryStream(document);
DeflateStream deflate = new DeflateStream(inStream, CompressionMode.Compress);
MemoryStream outStream = new MemoryStream();
int b;
while ((b = deflate.ReadByte()) != -1) {
    outStream.WriteByte((byte)b);
}
return outStream.ToArray();
```

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

Explanation: The document is compressed and written to a new MemoryStream using the Deflate class. Finally the compressed data can be returned as an array of bytes using the ToArray method of the MemoryStream.

A does not compress and write the document, instead it is compressing and writing an empty array

B & D are reading and writing to the same document.

· Leerer MemoryStream  
 · DeflateStream für MemoryStream und CompressionMode.Compress · mit deflateStream die übergebene Array (document) schreiben (was, ab wo, wie gross) · deflateStream schliessen (nun ist der komprimierte Inhalt im MemoryStream) · Inhalt des memoryStreams als ByteArray zurückliefern - mit MemoryStream.ToArray()

**QUESTION 17**

You are writing an application that uses SOAP to exchange data with other applications.

You use a Department class that inherits from ArrayList to send objects to another application.

The Department object is named dept.

You need to ensure that the application serializes the Department object for transport by using SOAP.

Which code should you use?

- A. 

```
SoapFormatter formatter = new SoapFormatter();
bytes[] buffer = new byte[dept.Capacity];
MemoryStream stream = new MemoryStream(buffer);
foreach (object o in dept) {
    formatter.Serialize(stream, o);
}
```
- B. 

```
SoapFormatter formatter = new SoapFormatter();
bytes[] buffer = new byte[dept.Capacity];
MemoryStream stream = new MemoryStream(buffer);
formatter.Serialize(stream, dept);
```



- C. SoapFormatter formatter = new SoapFormatter();  
MemoryStream stream = new MemoryStream();  
foreach (object o in dept) {  
formatter.Serialize(stream, o);  
}
- D. SoapFormatter formatter = new SoapFormatter();  
MemoryStream stream = new MemoryStream();  
formatter.Serialize(stream, dept);

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

Explanation: Simply serialize the entire object to a stream using a SoapFormatter.

A&C attempt to serialize components of the object rather the object itself.

B attempts to serialize to an array, however the array will not be big enough to store the serialized object because it is not sized on the entire object.

**QUESTION 18**

You are creating a class that performs complex financial calculations.

The class contains a method named GetCurrentRate that retrieves the current interest rate and a variable named currRate

that stores the current interest rate.

You write serialized representations of the class.

You need to write a code segment that updates the currRate variable with the current interest rate when an instance of the class is deserialized.

Which code segment should you use?

- A. [OnSerializing]  
internal void UpdateValue (StreamingContext context) {  
currRate = GetCurrentRate();  
}
- B. [OnSerializing]  
internal void UpdateValue(SerializationInfo info) {  
info.AddValue("currentRate", GetCurrentRate());  
}
- C. [OnDeserializing]  
internal void UpdateValue(SerializationInfo info) {  
info.AddValue("currentRate", GetCurrentRate());  
}
- D. [OnDeserialized]  
internal void UpdateValue(StreamingContext context) {  
currRate = GetCurrentRate();  
}

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

Explanation: A method with the OnDeserialized attribute will be called after Deserialization and any instance variables can be set.

A & B the method will fire during serializing, the question is concerned with reconstructing the object during

deserialization.

C the OnDeserializing attribute is useful for default values. OnDeserializing attribute works with a method that contains a StreamContext parameter and not a SerializationInfo parameter.

· Formulierung der Frage irreführend.

### QUESTION 19

You are writing an application that uses isolated storage to store user preferences.

The application uses multiple assemblies.

Multiple users will use this application on the same computer.

You need to create a directory named Preferences in the isolated storage area that is scoped to the current Microsoft Windows identity and assembly.

Which code segment should you use?

- A. IsolatedStrongFile store;  
store = IsolatedStorageFile.GetUserStoreForAssembly();  
store.CreateDirectory("Preferences");
- B. IsolatedStrongFile store;  
store = IsolatedStorageFile.GetStoreForAssembly();  
store.CreateDirectory("Preferences");
- C. IsolatedStrongFile store;  
store = IsolatedStorageFile.GetUserStoreForDomain();  
store.CreateDirectory("Preferences");
- D. IsolatedStrongFile store;  
store = IsolatedStorageFile.GetMachineStoreForApplication();  
store.CreateDirectory("Preferences");

**Answer:** A

**Section:** (none)

### Explanation/Reference:

The user store for the assembly is the correct store that is required. It is returned by IsolatedStorageFile.GetUserStoreForAssembly().

B,C & D return Isolated Storage File stores of incorrect scope

· Genau lesen: ...is scoped to the current Microsoft Windows identity (User) and assembly ...

· Also IsolatedStorageFile.GetUserStoreForAssembly()

· GetStoreForAssembly gibt es NICHT !!!

### QUESTION 20

You need to read the entire contents of a file named Message.txt into a single string variable.

Which code segment should you use?

- A. string result = null;  
StreamReader reader = new StreamReader("Message.txt");  
result = reader.Read().ToString();
- B. string result = null;  
StreamReader reader = new StreamReader("Message.txt");  
result = reader.ReadToEnd();

- C. `string result = string.Empty;`  
`StreamReader reader = new StreamReader("Message.txt");`  
`while (!reader.EndOfStream) {`  
`result += reader.ToString();`  
`}`
- D. `string result = null;`  
`StreamReader reader = new StreamReader("Message.txt");`  
`result = reader.ReadLine();`

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

Explanation: Create a StreamReader based on the file and call the ReadToEnd() method to quickly read the entire file and return a string.

A & D does not read the entire file.

C calling ToString() on the reader will give a string representation of the stream and will not read from the stream.

· ReadToEnd

· Read liest nur ein Zeichen !!!

**QUESTION 21**

You need to write a code segment that transfers the first 80 bytes from a stream variable named stream1 into a new byte array named byteArray.

You also need to ensure that the code segment assigns the number of bytes that are transferred to an integer variable named bytesTransferred.

Which code segment should you use?

- A. `bytesTransferred = stream1.Read(byteArray, 0, 80);`
- B. `for (int i = 0; i < 80; i++) {`  
`stream1.WriteByte(byteArray[i]);`  
`bytesTransferred = i;`  
`if (!stream1.CanWrite) {`  
`break;`  
`}`  
`}`
- C. `while (bytesTransferred < 80) {`  
`stream1.Seek(1, SeekOrigin.Current);`  
`byteArray[bytesTransferred++] = Convert.ToByte(stream1.ReadByte()); }`
- D. `stream1.Write(byteArray, 0, 80);`  
`bytesTransferred = byteArray.Length;`

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

Explanation: The Read() method accepts a byte array and the start position and number of bytes to read as parameters.

B & D The question indicates that data should be read from the stream not written to it.

C it is unnecessary to attempt to read byte by byte, the Read() method provides a very efficient way of reading into a byte array.

### QUESTION 22

You are writing a method that accepts a string parameter named message.

Your method must break the message parameter into individual lines of text and pass each line to a second method named Process.

Which code segment should you use?

- A. 

```
StreamReader reader = new StreamReader(message);
Process(reader.ReadToEnd());
reader.Close();
```
- B. 

```
StreamReader reader = new StreamReader(message);
while (reader.Peek() != -1) {
    string line = reader.Read().ToString();
    Process(line);
}
reader.Close();
```
- C. 

```
StreamReader reader = new StreamReader(message);
Process(reader.ToString());
reader.Close();
```
- D. 

```
StreamReader reader = new StreamReader(message);
while (reader.Peek() != -1) {
    Process(reader.ReadLine());
}
reader.Close();
```

**Answer:** D

**Section:** (none)

### Explanation/Reference:

Explanation: StreamReader.ReadLine() allows for lines to be read line by line.

A ReadToEnd() will read the entire stream.

B Read() will not read the line but only the next character.

C will not read from the message but will just give a string representation of the reader.

· ReadLine !!!

### QUESTION 23

You need to write a code segment that transfers the contents of a byte array named dataToSend by using a NetworkStream object named netStream.

You need to use a cache of size 8,192 bytes.

Which code segment should you use?

- A. 

```
MemoryStream memStream = new MemoryStream(8192);
memStream.Write(dataToSend, 0, (int) netStream.Length);
```
- B. 

```
MemoryStream memStream = new MemoryStream(8192);
netStream.Write(dataToSend, 0, (int) memStream.Length);
```
- C. 

```
BufferedStream bufStream = new BufferedStream(netStream, 8192);
bufStream.Write(dataToSend, 0, dataToSend.Length);
```

D. `BufferedStream bufStream = new BufferedStream(netStream);`  
`bufStream.Write(dataToSend, 0, 8192);`

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

Explanation: To send data using a cache it is necessary to use a `BufferedStream`. The `BufferedStream` should be created with the cache size of 8192 bytes.

A & B do not employ caching.

D does not correctly initialise the `BufferedStream` to have a cache size of 8192 bytes.

· `BufferedStream` und `Buffergröße` im Konstruktor angeben !!!

**QUESTION 24**

You need to serialize an object of type `List<int>` in a binary format.

The object is named `data`.

Which code segment should you use?

- A. `BinaryFormatter formatter = new BinaryFormatter();`  
`MemoryStream stream = new MemoryStream();`  
`formatter.Serialize(stream, data);`
- B. `BinaryFormatter formatter = new BinaryFormatter();`  
`MemoryStream stream = new MemoryStream();`  
`for (int i= 0;i <data.Count; i++) {`  
`formatter.Serialize(stream, data[i]);`  
`}`
- C. `BinaryFormatter formatter = new BinaryFormatter();`  
`byte[] buffer = new byte[data.Count];`  
`MemoryStream stream = new MemoryStream(buffer, true);`  
`formatter.Serialize(stream, data);`
- D. `BinaryFormatter formatter = new BinaryFormatter();`  
`MemoryStream stream = new MemoryStream();`  
`data.ForEach(delegate(int num){`  
`formatter.Serialize(stream, num);`  
`} );`

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

Explanation: create a `BinaryFormatter` and a `MemoryStream` and simply use the formatter to serialize the data to the stream.

B Collections support serialization, hence it is not required to try to serialize each item independently.

C The `MemoryStream` is created to be non resizeable and it is not the correct size.

**QUESTION 25**

Certkiller.com has given you the task of serializing an object and writing it to a data file using binary serialization.

You need to ensure that you meet these requirements.

What should you do?

- A. Use the following code:  
`object obj = new object();  
Stream objStream = File.Open("DataFile.dat", FileMode.Create);  
BinaryFormatter objFormatter = new BinaryFormatter ();  
objFormatter.Serialize(objStream, obj);`
- B. Use the following code:  
`object obj = new object();  
BinaryFormatter objFormatter = new BinaryFormatter ();  
objFormatter.Serialize(obj);`
- C. Use the following code:  
`Stream objStream = File.Open("DataFile.dat", FileMode.Create);  
BinaryFormatter objFormatter = new BinaryFormatter ();  
objFormatter.Serialize(objStream);`
- D. Use the following code:  
`object obj = new object();  
Stream objStream = File.Open ("DataFile.dat", FileMode.Create);  
BinaryFormatter objFormatter = new BinaryFormatter ();  
objFormatter.Serialize(obj, objStream);`

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

Explanation: This code instantiates an object named obj, opens a file stream, instantiates a BinaryFormatter object, and serializes the obj object to the DataFile.dat file. The File.Open method takes a file path string and FileMode enumeration value as arguments and returns a FileStream object. The Serialize method of the BinaryFormatter class takes two arguments: a stream and the object to be serialized. The Serialize method uses the stream to write the object to the destination.

Incorrect Answers:

B, C, D: If you use these options it will cause a compile-time error.

- ACHTUNG
- objFormatter.Serialize(objStream, obj); //RICHTIG
- NICHT
- objFormatter.Serialize(obj, objStream); //FALSCH

**QUESTION 26**

You have recently created a serializable class named Vehicle.

The class is shown below:

```
[Serializable]  
public class Vehicle{  
    public string VIN;  
    public string Make;  
    public string Model;  
    public string Year;  
}
```

You are planning to create a custom formatter class to control the formatting of Vehicle objects when they are serialized.

You need to ensure that is achieved with as little development effort as possible.

What should you do?

- A. Use the following code:  

```
public class VehicleFormatter : Formatter{
    //Implementation omitted
}
```
- B. Use the following code:  

```
public class VehicleFormatter : IGenericFormatter{
    //Implementation omitted
}
```
- C. Use the following code:  

```
public class VehicleFormatter : IFormatConverter{
    //Implementation omitted
}
```
- D. Use the following code  

```
public class VehicleFormatter : IFormatter{
    //Implementation omitted
}
```

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

When implementing the IFormatter interface, you must provide implementation for two methods and three properties.

The two methods, Serialize and Deserialize control how objects will be stored from memory and loaded into memory, respectively. Both methods accept a Stream object as an argument. The Serialize method also takes a generic object as its second argument and uses the Stream argument to write the object.

The Deserialize method returns the object using the Stream argument. The Binder, Context, and Surrogate Selector properties must also be implemented.

Incorrect Answers:

A: The Formatter class requires more development effort than necessary.

B: The IGenericFormatter interface does not exist.

C: The IFormatConverter interface does not control formatting during serialization.

· IFormatter !!! - nur die Methoden Serialize bzw. Deserialize · (Formatter wäre viel aufwendiger aufwendig - die ganzen WriteMethoden usw.)

**QUESTION 27**

You develop a serializable class for persisting objects as files.

Every time an object is serialized, you have to update the database with the name of the object and location of that file.

You elect to employ the OnSerialized attribute to achieve this objective.

You now need to apply the OnSerialized attribute to a certain method.

What should you do?

- A. Apply the OnSerialized attribute to the following method:  

```
public void AfterSerialization (object sender, EventArgs e){
    //Update database
}
```
- B. Apply the OnSerialized attribute to the following method:  

```
public void AfterSerialization (object sender){
    //Update database
}
```

- C. Apply the OnSerialized attribute to the following method:  
`public void AfterSerialization (StreamingContext context){  
 //Update database  
}`
- D. Apply the OnSerialized attribute to the following method:  
`public StreamingContext AfterSerialization (){  
 //Update database  
}`

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

This method contains the correct method signature to have the OnSerialized attribute applied. The method must accept a StreamingContext as an argument for accessing the read/write stream during serialization/deserialization.

Incorrect Answers:

A, B, D: You should not apply the OnSerialized attribute to these methods because they do not accept a StreamingContext as an argument.

· NUR StreamingContext !!!

**QUESTION 28**

You have recently created a serializable class named Vehicle.

The class is shown below:

```
[Serializable]  
public class Vehicle {  
    public string VIN;  
    public string Make;  
    private string Model;  
    private int Year;  
    private string Owner;  
}
```

Certkiller .com does not want the Owner field to be persisted when a Vehicle object is serialized, for security reasons.

You need to ensure that this objective is fulfilled.

What should you do?

- A. Apply the OptionalField attribute to the Owner field.
- B. Apply the **NonSerialized** attribute to the Owner field.
- C. Have the Vehicle class implement the IFormatter interface for custom serialization.
- D. Do nothing because, when using binary serialization, Private fields are never persisted.

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

This will ensure that the Owner field will not be serialized, but it will allow all other fields to be serialized normally.

Incorrect Answers:

A: This option would be used for deserialization.

C: This option would require excessive developer effort.

D: This is incorrect because all fields marked private or otherwise are persisted when using binary serialization.



· NonSerialized !!!

### QUESTION 29

You have recently written the code shown below:

```
Hashtable emailAddresses = new Hashtable ();
emailAddresses.Add ("Mia", "mia@ Certkiller .com");
emailAddresses.Add ("Andy", "andy@ Certkiller .com");
emailAddresses.Add ("Kara", "kara@ Certkiller .com");
FileStream stream = new FileStream ("Email.dat", FileMode.Create);
BinaryFormatter formatter = new BinaryFormatter ();
formatter.Serialize (stream, emailAddresses);
```

You need to ensure that you are able to load the emailAddresses object from the Email.dat file into your application.

What should you do?

- A. Use the following code:  

```
FileStream readStream = new FileStream("Email.dat", FileMode.Open);
HashTable loadEmails = readStream.Deserialize();
```
- B. Use the following code:  

```
FileStream readStream = new FileStream ("Email.dat", FileMode.Open);
BinaryFormatter readFormatter = new BinaryFormatter();
HashTable loadEmails = readFormatter.Deserialize(readStream);
```
- C. Use the following code:  

```
FileStream readStream = new FileStream("Email.dat", FileMode.Open);
BinaryFormatter readFormatter = new BinaryFormatter ();
HashTable loadEmails = (HashTable)readFormatter.Deserialize(readStream);
```
- D. Use the following code:  

```
FileStream readStream = new FileStream ("Email.dat", FileMode.Open);
HashTable loadEmails = (HashTable)readFormatter.ReadObject ();
```

**Answer:** C

**Section:** (none)

#### Explanation/Reference:

Explanation: This instantiates a BinaryFormatter object, and deserializes the emailAddresses object from the Email.dat file.

The FileStream constructor takes a file path string and FileMode enumeration value as arguments.

The Deserialize method of the BinaryFormatter class takes the stream of the object to be deserialized and returns a generic object.

This generic object must be cast or converted to the HashTable data type.

Incorrect Answers:

A, D: You should not use the code fragments that do not instantiate the BinaryFormatter object because the ReadObject and Deserialize methods do not exist in the FileStream class.

B: You should not use the code that does not cast or convert the return value of the Deserialize method because the Deserialize method returns a generic object.

· BinaryFormatter und CASTING !!!

### QUESTION 30

you have recently written the code shown below:

```
Hashtable emailAddresses = new Hashtable ();
emailAddresses.Add ("Mia", "mia@ Certkiller .com");
emailAddresses.Add ("Andy", "andy@ Certkiller .com");
emailAddresses.Add ("Kara", "kara@ Certkiller .com");
```

You need to ensure that these e-mail addresses are stored in the Email.dat file so that you can load them again when the user restarts the application.

What should you do?

- A. Add the following code:  

```
FileStream stream = new FileStream ("Email.dat", FileMode.Create);
BinaryFormatter formatter = new BinaryFormatter ();
formatter.Deserialize(stream, emailAddresses);
```
- B. Add the following code:  

```
FileStream stream = new FileStream ("Email.dat", FileMode.Create);
BinaryFormatter formatter = new BinaryFormatter ();
formatter.Serialize(stream, emailAddresses);
```
- C. Add the following code:  

```
FileStream stream = new FileStream ("Email.dat", FileMode.Create);
stream.Serialize(emailAddresses);
```
- D. Add the following code:  

```
FileStream stream = new FileStream ("Email.dat", FileMode.Create);
stream.WriteObject(emailAddresses);
```

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

This code instantiates a file stream, instantiates a BinaryFormatter object, and serializes the emailAddresses object to the Email.dat file. The FileStream constructor takes a file path string and FileMode enumeration as arguments. The Serialize method of the BinaryFormatter class takes two arguments, a stream and the object to be serialized.

The Serialize method uses the stream to write the object to the destination.

Incorrect Answers:

A: You should not add the code that invokes the Deserialize method of the BinaryFormatter class because you must serialize the object first. C D: You should not add the code fragments that do not instantiate the BinaryFormatter object because the WriteObject and Serialize methods do not exist in the FileStream class.

· Achtung die Frage meint Serialisierung und nicht Kontrolle mit Deserialisierung · dann ist es einfach oder !!!

**QUESTION 31**

You have recently completely creating a new application for Certkiller .com.

This new application has to load an instance of the Inventory class from a large file named Inventory.dat.

You need to ensure that the application executes the loading process in as little time as possible.

What should you do?

- A. Use the following code:  

```
FileStream readStream = new FileStream ("Inventory.dat", FileMode.Open);
BinaryFormatter readFormatter = new BinaryFormatter ();
Inventory currentInventory = (Inventory) readFormatter.FastDeserialize(readStream);
```

- B. Use the following code:  
`FileStream readStream = new FileStream ("Inventory.dat", FileMode.Open);  
 BinaryFormatter readFormatter = new BinaryFormatter ();  
 Inventory currentInventory = (Inventory)readFormatter.Deserialize (readStream);`
- C. Use the following code:  
`FileStream readStream = new FileStream ("Inventory.dat", FileMode.Open);  
 BinaryFormatter readFormatter = new BinaryFormatter ();  
 Inventory currentInventory = (Inventory)readFormatter.UnsafeDeserialize(readStream);`
- D. Use the following code:  
`FileStream readStream = new FileStream ("Inventory.dat", FileMode.Open);  
 BinaryFormatter readFormatter = new BinaryFormatter ();  
 Inventory currentInventory = (Inventory)readFormatter.SafeDeserialize(readStream);`

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

This code instantiates a file stream, instantiates a BinaryFormatter object, and deserializes an Inventory object to the Inventory.dat file.

The UnsafeDeserialize and Deserialize methods perform the same operation, but the UnsafeDeserialize method uses unmanaged code and requires more permission.

Because the UnsafeDeserialize method uses unmanaged code, your code should be granted full trust to execute properly.

The UnsafeDeserialize method of the BinaryFormatter class takes two arguments, the stream of the object to be deserialized

and the HeaderHandler object to deal with any binary headers. The UnsafeDeserialize method returns a generic object that must be cast or converted to the Inventory data type.

Incorrect Answers:

A, D: You should not use the code that invokes the FastDeserialize and SafeDeserialize methods because no such methods exist for the BinaryFormatter class.

B: You should not use the code that invokes the Deserialize method because the UnsafeDeserialize method yields better performance.

· UnsafeDeserialize - Methode mit Casting !!! SCHNELLER als Deserialize (unmanaged code !!!)

**QUESTION 32**

Certkiller .com has asked you to develop an application that displays the properties for all Certkiller.com's network drives.

The information generated by this application will be utilized by Certkiller .com's network administrators to verify client setups.

You need to ensure that these requirements are fully satisfied.

What should you do?

- A. Use the following code:  
`public void EnumerateNetworkDrives (){  
 foreach (Drive netDrive in Drive.GetDrives()){  
 if (netDrive.DriveType == DriveType.Network)  
 Console.WriteLine ("{0} ({1}) : {2} bytes", netDrive.Name, netDrive.VolumeLabel,netDrive.TotalSize);  
 }  
 }  
 }`

- B. Use the following code:
- ```
public void EnumerateNetworkDrives (){
    foreach (DriveInfo netDrive in DriveInfo.GetDrives()){
        Console.WriteLine ("{0} ({1}) : {2} bytes", netDrive.Name, netDrive.VolumeLabel,netDrive.TotalSize);
    }
}
```
- C. Use the following code:
- ```
public void EnumerateNetworkDrives (){
    foreach (DriveInfo netDrive in DriveInfo.GetDrives()){
        if (netDrive.DriveType == DriveType.Network)
            Console.WriteLine ("{0} ({1}) : {2} bytes", netDrive.Name, netDrive.VolumeLabel,netDrive.TotalSize);
    }
}
```
- D. Use the following code:
- ```
public void EnumerateNetworkDrives (){
    foreach (DriveInfo netDrive in DriveInfo.GetDrives(DriveType.Network)){
        Console.WriteLine ("{0} ({1}) : {2} bytes", netDrive.Name, netDrive.VolumeLabel,netDrive.TotalSize);
    }
}
```

**Answer:** C

**Section:** (none)

#### Explanation/Reference:

The GetDrives method is invoked and returns an array of DriveInfo objects representing the available drives. The DriveType property is used to verify that the netDrive variable is a network drive. The DriveType property returns a DriveType enumeration value that can be Network, Fixed, CDRom, or Removable. If the DriveType property is DriveType.Network, then the Console.WriteLine method displays the Name, VolumeLabel and TotalSize properties of the DriveInfo object to the console.

Incorrect Answers:

A: You should not use the code that uses the Disk class because there is no such class in the System.IO namespace.

B D: You should not use the code fragments that do not retrieve the DriveType property of the DriveInfo object because there is no such signature for the GetDrives method and no such method as GetNetworkDrives.

· public static DriveInfo[] GetDrives () !!! (keine Parameter möglich !!!)

#### QUESTION 33

Certkiller.com has asked you to create an application to display all of the top directories based on the drive path.

You need to ensure that the application displays the number of files within top-level directories. What should you do?

- A. Use the following code:
- ```
public void DisplayDriveDirectories (string drivePath)
{ if (Directory.Exists(drivePath)){
    foreach (String dirPath in Directory.GetDirectories(drivePath)){
        DirectoryInfo dir = new DirectoryInfo (drivePath);
        int numFiles = dir.TotalFiles;
        Console.WriteLine( "{0} : {1} files.", dir.Name, numFiles); }
    }
}
```

B. Use the following code:

```
public void DisplayDriveDirectories (string drivePath)
{ if (Directory.Exists (drivePath)){
  foreach (String dirPath in Directory.GetDirectories (drivePath)){
    DirectoryInfo dir = new DirectoryInfo(drivePath);
    int numFiles = dir.Length;
    Console.WriteLine( "{0} : {1} files.", dir.Name, numFiles); }
  }
}
```

C. Use the following code:

```
public void DisplayDriveDirectories (string drivePath)
{ if (Directory.Exists(drivePath)){
  foreach (String dirPath in Directory.GetDirectories(drivePath)){
    DirectoryInfo dir = new DirectoryInfo(drivePath);
    int numFiles = dir.GetFiles().Length;
    Console.WriteLine( "{0} : {1} files.", dir.Name, numFiles); }
  }
}
```

D. Use the following code:

```
public void DisplayDriveDirectories (string drivePath)
{ if (Directory.Exists(drivePath)){
  foreach (String dirPath in Directory.GetDirectories (drivePath)){
    DirectoryInfo dir = new DirectoryInfo (drivePath);
    int numFiles = dir.Size;
    Console.WriteLine( "{0} : {1} files.", dir.Name, numFiles); }
  }
}
```

**Answer: C**

**Section:** (none)

#### **Explanation/Reference:**

Explanation: This code iterates through each top level of a given drive path and displays the Name property and number of files.

First, the Exists method verifies that the drive path exists. Then, the GetDirectories method is invoked and returns a string array of directory paths.

GetDirectories takes a directory path as an argument.

Then a DirectoryInfo object is instantiated using the dirPath variable as it is updated with each iteration.

The number of files in the directory is evaluated by using the GetFiles method, which returns an array of FileInfo objects and retrieves the Length Property of the array.

The value is assigned to the numFiles variable.

The DirectoryInfo object represents metadata about a directory instance. The Console.WriteLine method displays the Name property of the DirectoryInfo object and the numFiles variable.

Incorrect Answers:

A, B, D: The TotalFiles, Length, and Size properties do not exist in the DirectoryInfo class.

· directoryInfo.GetFiles().Length !!!

#### **QUESTION 34**

Certkiller .com has asked you to create a file management application to monitor the hosts file.

Certkiller .com has instructed you to change the hosts file if it has been changed.

You, therefore, need to display the size and whether the hosts file is set to read-only.

What should you do?

- A. Use the following code:  
**FileInfo** hosts = new **FileInfo**("C:\\Windows\\system32\\drivers\\etc\\hosts");  
 Console.WriteLine ("ReadOnly?" + hosts.**IsReadOnly**);  
 Console.WriteLine ("Size?" + **hosts.Length**);
- B. Use the following code:  
 File hosts = new File(@"C:\\Windows\\system32\\drivers\\etc\\hosts");  
 Console.WriteLine ("ReadOnly?" + hosts.**IsReadOnly**);  
 Console.WriteLine ("Size?" + hosts.**Length**);
- C. Use the following code:  
 File hosts = new File(@"C:\\Windows\\system32\\drivers\\etc\\hosts");  
 Console.WriteLine ("ReadOnly?" + hosts.**GetReadOnly**);  
 Console.WriteLine ("Size?" + hosts.**GetLength**);
- D. Use the following code:  
 FileInfo hosts = new FileInfo(@"C:\\Windows\\system32\\drivers\\etc\\hosts");  
 Console.WriteLine ("ReadOnly?" + hosts.**IsReadOnly**);  
 Console.WriteLine ("Size?" + hosts.**Size**);

**Answer:** A

**Section:** (none)

#### **Explanation/Reference:**

Explanation: This code instantiates a FileInfo object using a file path string and outputs the IsReadOnly and Length properties to the command line. The FileInfo object represents the information about a system file. The FileInfo class contains common properties and methods for reading and setting file metadata and contents. The IsReadOnly property returns a Boolean value indicating whether the file is set to read- only. The Length property returns the size of the file in bytes.

Incorrect Answers:

B C: You should not use the code that instantiate a File object because the File class does not contain an IsReadOnly or Length property and is a static class. Also, you should not use the code fragments that specify the Size property because no such property exists.

D: You should not use the code fragments that specify the Size method because no such methods exist.

· FileInfo IsReadOnly und Length Properties

#### **QUESTION 35**

You have recently completed the creation of a new application.

Certkiller .com requires you to ensure that this new application creates a file that contains an array of bytes. What should you do?

- A. Use the following code:  

```
public void WriteBytes (byte [] bytes){
  FileStream fs = new FileStream ("C:\\file.txt", FileMode.Create);
  for (int i = 0; i < bytes.Length - 1; i++)
    fs.Write(bytes [i]);
  fs.Close ();
}
```
- B. Use the following code:  

```
public void WriteBytes (byte [] bytes){
  FileStream fs = new FileStream ("C:\\file.txt", FileMode.Create);
  for (int i = 0; i < bytes.Length - 1; i++)
    fs.WriteByte(bytes [i]);
  fs.Close ();
}
```

- C. Use the following code:
- ```
public void WriteBytes (byte [] bytes){
    FileStream fs = new FileStream ("C:\\file.txt", FileMode.Create);
    fs.WriteByte(bytes, 0, bytes.Length);
    fs.Close ();
}
```
- D. Use the following code:
- ```
public void WriteBytes (byte [] bytes){
    FileStream fs = new FileStream ("C:\\file.txt", FileMode.Create);
    fs.Write(bytes, 0, bytes.Length);
    fs.Close ();
}
```

**Answer:** C

**Section:** (none)

### Explanation/Reference:

Answer D

The FileStream constructor accepts a string argument as the file path and a FileMode enumeration value.

The FileMode enumeration value indicates the file stream will be used, and includes the values Append, Create, CreateNew, Open, and Truncate.

The FileMode.Create value indicates a new file will be created or, if one already exists, that it will be overwritten.

The FileStream class includes a Write method for writing an array of bytes.

The Write method takes a byte array, offsetvalue and total number of bytes as arguments.

The other method, WriteByte, takes a single argument of the type of byte, and it requires manual iteration to write an array.

Like all streams, the FileStream object has a Close method, which should be called after work is done with the stream.

Incorrect Answers:

A, C: You should not use the code fragments that invokes the Write method or the WriteBytes method with only one argument because no such signatures exists in the FileStream class.

B: This code is unnecessary because the FileStream class also contains a Write method that takes a byte array as an argument.

· Achtung: Es gibt nur

· fileStream.Write(ByteArray, Startindex, Anzahl);

· und

· fileStream.WriteByte(byte myByte); // nur ein Byte !!!

### QUESTION 36

You are currently in the process of creating an application that reads binary information from a file.

You need to ensure that the only the first kilobyte of data is retrieved.

What should you do?

- A. Use the following code:
- ```
FileStream fs = new FileStream("C:\\file.txt", FileMode.Open);
BufferedStream bs = new BufferedStream (fs);
byte [ ] bytes = new byte [1023];
bs.Read (bytes, 0, bytes.Length);
bs.Close ();
for (int i = 0; i < bytes.Length-1; i++)
Console.WriteLine ("{0} : {1}", i, bytes [i]);
```
- B. Use the following code:
- ```
FileStream fs = new FileStream("C:\\file.txt", FileMode.Open);
byte [ ] bytes = new byte [1023];
fs.Read (bytes, 0, bytes.Length);
fs.Close ();
for (int i = 0; i < bytes.Length-1; i++)
Console.WriteLine ("{0} : {1}", i, bytes [i]);
```
- C. Use the following code:
- ```
FileStream fs = new FileStream("C:\\file.txt", FileMode.Open);
BufferedStream bs = new BufferedStream (fs);
byte [ ] bytes = new byte [1023];
bytes = bs.ReadAllBytes (0, 1023);
bs.Close ();
for (int i = 0; i < bytes.Length-1; i++)
Console.WriteLine ("{0} : {1}", i, bytes [i]);
```
- D. Use the following code:
- ```
FileStream fs = new FileStream("C:\\file.txt", FileMode.Open);
BufferedStream bs = new BufferedStream (fs);
byte [ ] bytes = new byte [1023];
bs.Read (bytes);
bs.Close ();
for (int i = 0; i < bytes.Length-1; i++)
Console.WriteLine ("{0} : {1}", i, bytes [i]);
```

**Answer: B**

**Section: (none)**

#### **Explanation/Reference:**

The FileStream constructor accepts a string argument as the file path and a FileMode enumeration value. The FileMode enumeration value indicates the file stream will be used, and includes the values Append, Create, CreateNew, Open, and Truncate.

The FileMode.Open value indicates a file will be opened if existing, or else a FileNotFoundException object will be thrown.

An array of 1024 bytes is instantiated.

The Read method takes the byte array, offset value, and total number of bytes as arguments. The other method ReadByte returns a single byte at a time, but it requires manual iteration to write an array.

Like all streams, the FileStream object has a Close method, which should be called after work is done with the stream.

The Console.WriteLine method is invoked to display the byte index and byte value.

**Incorrect Answers:**

A, C, D: You should not use the code that specifies a buffered stream BECAUSE the FileStream class IS ALREADY a buffered stream. Also, you should not use the code fragments that invoke the ReadAllBytes method and

the Read method with the incorrect arguments because no such method signatures exist.

· FileStream ist ein BufferedStream und kann daher direkt verwendet werden !!!



### QUESTION 37

You are in the process of creating a new application.  
This new application has to be able to read all data from a text file.  
What should you do?

- A. Use the following code:  

```
FileStream fs = new FileStream("C:\\file.txt", FileMode.Open);  
StreamReader sr = new StreamReader (fs);  
string data = sr.ReadToEnd ();  
sr.Close ();  
Console.WriteLine (data);
```
- B. Use the following code:  

```
FileStream fs = new FileStream("C:\\file.txt", FileMode.Open);  
string data = sr.ReadToEnd ();  
fs.Close ();  
Console.WriteLine (data);
```
- C. Use the following code:  

```
FileStream fs = new FileStream("C:\\file.txt", FileMode.Open);  
StringBuilder data = new StringBuilder();  
string data;  
while (sr.Peek () > -1)  
data += sr.ReadLine ();  
sr.Close ();  
Console.WriteLine (data);
```
- D. Use the following code:  

```
FileStream fs = new FileStream("C:\\file.txt", FileMode.Open);  
StreamReader sr = new StreamReader (fs);  
StringBuilder data = new StringBuilder ();  
while (sr.Peek () > -1)  
data.Append (sr.ReadLine ());  
sr.Close ();  
Console.WriteLine (data.ToString ());
```

**Answer:** A

**Section:** (none)

#### Explanation/Reference:

The FileStream constructor accepts a string argument as the file path and a FileMode enumeration value.

The FileMode enumeration value indicates the file stream will be used, and includes the values

Append, Create, CreateNew, Open, and Truncate.

The FileMode.Open value indicates a file will be opened if existing, or else a FileNotFoundException object will be thrown.

A StreamReader object is instantiated using the FileStream object as input. The ReadToEnd method returns a string representing all data from that position to the end of the file.

There are two other read methods, ReadLine and ReadBlock. The ReadLine method returns a string representing all data from that position to the end of a line return.

The ReadBlock method takes a character array, offset value and total number of characters as arguments.

Like all streams, the StreamReader object has a Close method, which should be called after work is done with the stream.

The Console.WriteLine method is invoked to display the data to the console.

**Incorrect Answers:**

B: You should not use the code that does not specify the StreamReader class because the FileStream class does not contain a ReadToEnd method.

C: You should not use the code that specifies a string object when invoking the ReadLine method rather than a StringBuilder object. The string object is less efficient than StringBuilder objects when performing concatenation operations.

D: This code should not be used because it manually iterates through the file using the ReadLine method, whereas the ReadToEnd method is more efficient.

· StreamReader und StreamReader.Close() - schliesst den verwendeten FileStream.

### QUESTION 38

You work as an application developer at Certkiller .com. You are currently in the process of creating a new application for Certkiller .com. You are required to read compressed data files that has been sent by Certkiller .com's sales offices.

These data files are less than 4 GB in size, but was compressed without cyclic redundancy. You want to write a method that receives the compressed files and return the uncompressed data as a byte array.

What should you do?

A. Use the following code:

```
public byte [] DecompressFile(string file){  
    FileStream fs = new FileStream(file, FileMode.Open);  
    DeflateStream cs = new DeflateStream(fs, CompressionMode.Decompress, true);  
    byte [ ] data = new byte [fs.Length - 1];  
    cs.Read(data, 0, data.Length);  
    cs.Close ();  
    return data;  
}
```

B. Use the following code:

```
public byte [] DecompressFile(string file){  
    FileStream fs = new FileStream(file, FileMode.Open);  
    GZipStream cs = new GZipStream(fs, CompressionMode.Decompress);  
    byte [ ] data = new byte [fs.Length - 1];  
    cs.Read(data, 0, data.Length);  
    return data;  
}
```

C. Use the following code:

```
public byte [] DecompressFile(string file){  
    FileStream fs = new FileStream(file, FileMode.Open);  
    DeflateStream cs = new DeflateStream (fs, CompressionMode.Decompress);  
    byte [ ] data = new byte [fs.Length - 1];  
    cs.Read(data, 0, data.Length);  
    return data;  
}
```

D. Use the following code:

```
public byte [] DecompressFile (string file){  
    FileStream fs = new FileStream (file, FileMode.Open);  
    GZipStream cs = new GZipStream (fs, CompressionMode.Decompress, true);  
    byte [ ] data = new byte [fs.Length - 1];  
    cs.Read (data, 0, data.Length);  
    cs.Close ();  
    return data;  
}
```

**Answer:** A

**Section:** (none)

### Explanation/Reference:

The DeflateStream uses the LZ77 and Huffman coding algorithms for lossless compression and decompression without cyclic redundancy.

The DeflateStream constructor takes a stream (in this case an input stream), a CompressionMode enumeration value,

and a Boolean value indicating whether to keep the stream open. The CompressionMode enumeration value indicates whether to compress or decompress the specified stream using the values Compress and Decompress, respectively. The Read method takes the byte array, offset value, and total number of bytes as arguments. In this code, the compressed data is read from the FileStream object associated with the DeflateStream object, and it is stored in the byte array.

Like all streams, the DeflateStream object has a Close method, which should be called after work is done with the stream.

Finally, the resulting byte array is returned.

Incorrect Answers:

B, D: You should not use the code fragments that specify the GZipStream class because this data format includes a data corruption check during decompression.

C: You should not use the code fragments that instantiate the Stream objects and invoke the Read method with the wrong arguments.

- DeflateStream (NO data corruption check)
- CompressionMode.Decompress bzw. CompressionMode.Compress · Read-Methode bzw. Write-Methode
- Warum ist C fasch ??? - kein Close !!!

### QUESTION 39

Certkiller .com has a file server named **Certkiller -SR07** that stores old inventory files. Certkiller .com has given you the task of creating an application to archive these old inventory files.

The inventory files have to be compressed prior to being uploaded to Certkiller .com's Web server.

You are currently writing a method that will receive a byte array and compress it into a new file.

You need to ensure that a data corruption check takes place during the decompression process.

What should you do?

A. Use the following code:

```
public void CompressFileWrite(string file, byte[] data){
    FileStream fs = new FileStream(file, FileMode.Create);
    DeflateStream cs = new DeflateStream( fs, Compressionmode.Compress, true);
    cs.Write (data, 0, data.Length);
    cs.Close();
}
```

B. Use the following code:

```
public void CompressFileWrite(string file, byte[] data){
    FileStream fs = new FileStream (file, FileMode.Create);
    GZipStream cs = new GZipStream( fs, Compressionmode.Compress, true);
    cs.Compress (data, 0, data.Length);
    cs.Close ();
}
```

C. Use the following code:

```
public void CompressFileWrite(string file, byte[] data){
    FileStream fs = new FileStream(file, FileMode.Create);
    DeflateStream cs = new DeflateStream( fs, Compressionmode.Compress, true);
    cs.Compress(data, 0, data.Length);
    cs.Close ();
}
```

D. Use the following code:

```
public void CompressFileWrite (string file, byte[] data){  
    FileStream fs = new FileStream(file, FileMode.Create);  
    GZipStream cs = new GZipStream( fs, Compressionmode.Compress, true);  
    cs.Write(data, 0, data.Length);  
    cs.Close();  
}
```

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

Explanation:

Incorrect Answers:

A, B, C: You should not use the code fragments that specify the DeflateStream class because this data format does not ensure that a data corruption check occurs during decompression. You should also not use the code that invokes the Compress method because no such method exists in the GZipStream or the DeflateStream classes.

- GZIP HAT data corruption check
- CompressionMode.Decompress bzw. CompressionMode.Compress · Read-Methode bzw. Write-Methode

**QUESTION 40**

You define the following regular expression of currency values:

```
regex tx = new regex("^-?\d+(\.\d{2})?$$")
```

You are required to write code that will be used to find whether a string in the variable named Bill matches the regular expression or not. You are also required to use this code as the expression in a conditional statement and need to know which code segment to use.

What should you do?

- A. The tx.Matches(Bill)
- B. The tx.Equals(Bill)
- C. The tx.Match(Bill)
- D. The tx.IsMatch(Bill)

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

In order for you to successfully indicate whether the regular expression finds a match in the input string you should make use of the IsMatch(Bill) segment in the scenario.

Incorrect Answers:

A: This segment should not be used in the scenario because the Matches method is used to search an input string for all occurrences of a regular expression and returns all the successful matches.

B: This segment should not be used in the scenario because this method is used to determine whether any two Object instances are equal.

C: This segment should not be used in the scenario because the Match method is used to search an input string for an occurrence of a regular expression and returns the precise results as a single successful match.

- IsMatch(...) liefert True bzw. False !!!

#### QUESTION 41

You work as the application developer at Certkiller .com.  
Certkiller .com uses Visual Studio.NET 2005 as its application development platform.  
You are developing a .NET Framework 2.0 application that will be used for

geographical mapping. Whenever you load a plug-in you create a separate application domain.  
You want to specify a list of directories that are searched for private assemblies.  
You are required to add the application's base directory as part of your search and need to configure an application domain to meet the requirements.

- A. The AppDomainSetup.PrivateBinPathProbe property should be used.
- B. The AppDomain.BaseDirectory property should be used.
- C. The AppDomain.DynamicDirectory property should be used.
- D. The AppDomainSetup.PrivateBinPath property should be used.

**Answer:** D

**Section:** (none)

#### Explanation/Reference:

Explanation: The method used in the answer is used to specify a list of directories under the application's base directory that are probed for private assemblies. This property must be set to specify the locations that should be searched.

Incorrect Answers:

A: This property is set to a non-empty value and excludes the ApplicationBase from the searches and should not be used.

B: This method only specifies the base directory for the application domain and does not specify all the different locations to be searched for private assemblies.

C: This property is used to get the directory that the assembly resolver uses to probe for dynamically created assemblies and is a read-only assembly.

#### QUESTION 42

You work as the application developer at Certkiller .com. Certkiller .com uses Visual Studio.NET 2005 as its application development platform.  
You are developing a .NET Framework 2.0 application class library and decide to use the AppDomainSetup class to create an application domain.  
You are required to create an instance of a type named BillSpace.BillType.  
You are additionally required to configure the new application domain so that it probes for the assembly containing the type in a specified directory.  
What should you do?

- A. The ApplicationBase property of the AppDomainSetup class should be used.
- B. The ActivationArguments property of the AppDomainSetup class should be used.
- C. The CachePath property of the AppDomainSetup class should be used.
- D. The AppDomainInitializerArguments property of the AppDomainSetup class should be used.

**Answer:** A

**Section:** (none)

#### Explanation/Reference:

Explanation: It is possible to provide the common language runtime with configuration information for a new application domain using the AppDomainSetup class. The most important property is the ApplicationBase when creating your own

application domains which is used to define the root directory of the application.

Incorrect Answers:

B: This method is incorrect and should not be used because it sets or gets data about activation of an application domain.

C: This method is incorrect and should not be used because it sets or gets the name of an area specific to the application where files are shadow copied.

D: This method is incorrect and should not be used because it sets or gets AppDomainInitializer delegate which represents a callback method.

#### QUESTION 43

You work as the application developer at Certkiller .com. Certkiller .com uses Visual Studio.NET 2005 as its application development platform.

You are developing a Windows Forms application. You are required to **create a new class** that raises a large number of events and keep memory usage of the class as low as possible.

This task must be accomplished using the minimum amount of code possible.

What should you do?

- A. **One instance of** the EventHandlerList class should be **used to store the delegate** defined for each event in the class
- B. One instance of the EventInstance class should be used for each event in the class
- C. The EventBuilder class should be used to define the events for the class
- D. One member variable per event delegate should be created per event delegate instance in the class

**Answer: A**

**Section:** (none)

#### Explanation/Reference:

Explanation: The best option in this scenario would be to make use of the EventHandlerList class since the number of events is large and this method provides a memory efficient mechanism for storing a list of delegates.

Incorrect Answers:

B: This method is used for representing information for an event log entry.

C: This method is used for defining events for a dynamically generated class and in the scenario the class is not dynamically generated.

D: This method should not be used as it is not a memory efficient solution which is what is required of you.

#### QUESTION 44

You work as the application developer at Certkiller .com.

You are developing a .NET Framework 2.0 Windows Service application named TestScannerSync that will be used to constantly monitor for a scanner to be connected to the computer.

The TestScannerSync class inherits its functionality from the ServiceBase class.

When you connect a scanner the service synchronizes its data with the data in a SQL Server database.

You decide to override the OnStop method in the TestScannerSync class and write code to close database connections.

You are required to identify a place in the code where you can open the connection to the database.

What should you do?

- A. The database connection is opened in the constructor of TestScannerSync class
- B. The database connection is opened in the OnContinue method of TestScannerSync class
- C. The database connection is opened in the **OnStart method** of the TestScannerSync class
- D. The database connection is opened in the OnPowerEvent method of the TestScannerSync class

**Answer: C**

**Section:** (none)

**Explanation/Reference:**

Explanation: The constructor for a service class derived from ServiceBase is called the first time you call Start on the service, the OnStart command-handling method is called immediately after the constructor executes.

Incorrect Answers:

A: The constructor is the incorrect choice as this is only called upon once and not again.

B: This method should not be used as this code will be executed when a Continue command is sent to a service in a paused state.

D: This method should not be considered as this method will execute when the computer's power state has changed.

**QUESTION 45**

You work as the application developer at Certkiller .com. Certkiller .com uses Visual Studio.NET 2005 as its application development platform.

You are developing an application that will be used to connect and control the behavior of existing services installed on a network server named Certkiller -SR01.

What should you do?

- A. Use the **MachineName** and ServiceName properties of the ServiceController class.
- B. Use the Site property of the ServiceController class.
- C. Use the Site property of the ServiceInstaller class.
- D. Use the ServiceName and DisplayName properties of the ServiceInstaller class.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

Explanation: The proper way to connect and control the behavior of existing services is by using the ServiceController class after which you are required to set two properties on it to identify the service to interact with. The MachineName property is used to define the computer Certkiller -SR01.

Incorrect Answers:

B, C: The Site property of the ServiceController and ServiceInstaller class should not be used because they bind a component to a container and enables communication between them.

D: This property should not be used as it is meant to specify the name of the service at the time of installation. The DisplayName property is used to specify the friendly name of the service at the time of installation.

**QUESTION 46**

You work as the application developer at Certkiller .com. Certkiller .com uses Visual Studio.NET 2005 as its application development platform.

You are developing a Windows Service application that consists of two Services.

The first service monitors a directory for new orders while the other service replicates a database table with up-to-date information.

You are required to develop a project installer class to install these services.

What should you do? (Choose two)

- A. **One ServiceProcess**Installer instance must be Instantiate and add it to the project installer class
- B. **Two ServiceInstaller** instances must be Instantiate and add them to the project installer class
- C. Two ServiceProcessInstaller instances must be Instantiate and add them to the project installer class
- D. One ComponentInstaller instance must be Instantiate and add them to the project installer class
- E. Two ComponentInstaller instances must be Instantiate and add them to the project installer class
- F. One ServiceInstaller instance must be Instantiate and add them to the project installer Class

**Answer:** AB

**Section:** (none)

**Explanation/Reference:**

Explanation: The proper way to do what the scenario requires of you would be to create one ServiceProcessInstaller instance per service application and one ServiceInstaller instance for each service in the application which have to be Instantiate.

Incorrect Answers:

C, F: You must create one ServiceProcessInstaller instance per service application and one ServiceInstaller instance for each service in the application which have to be Instantiate.

D, E: The ComponentInstaller class should not be considered for use as this class is an abstract class that can not be Instantiate.

**QUESTION 47**

You work as the application developer at Certkiller .com. Certkiller .com uses Visual Studio.NET 2005 as its application development platform.

You are developing a Windows Service application. You need to change the security context in which the service runs.

The service will be run in context of a non-privileged user on the local computer and present anonymous credentials to any remote server.

You are required to set the Account property of the ServiceProcessInstaller class to specify the service account.

What should you do?

- A. Use ServiceAccount.User.
- B. Use ServiceAccount.LocalService.
- C. Use ServiceAccount.LocalSystem.
- D. Use ServiceAccount.NetworkService.

**Answer: B**

**Section: (none)**

**Explanation/Reference:**

Explanation: The proper way to complete the operation is to use the ServiceAccount.LocalService as the value of the Account property of the ServiceProcessInstaller class, setting this value will run the service in context of the non-privileged user on the local computer.

Incorrect Answers:

A: This method should not be used as you will be specifying the service to run in the security context of a specified user account.

C: This method should not be considered as you will run the service in the security context of a highly privileged user account.

D: The scenario requires the service to run in the security context of a non-privileged user on the local system not the network.

**QUESTION 48**

You work as the application developer at Certkiller .com. Certkiller .com uses Visual Studio.NET 2005 as its application development platform.

You are developing a Windows Service application which contains three different Windows services.

You are required to only set one Windows service to start automatically when the system is restarted.

What should you do?

- A. Use the ServiceController class.
- B. Use the ServiceBase class.
- C. Use the ServiceProcessInstaller class.
- D. Use the ServiceInstaller class.



**Answer:** D

**Section:** (none)

**Explanation/Reference:**

Explanation: The proper and best way to achieve the scenario would be to use the ServiceInstaller class, the StartType property of the ServiceInstaller class allows you to specify how and when a service is started.

Incorrect Answers:

A: This method used here is incorrect as it can not be used to specify the start type of a Windows service.

B: This method is the base type for all Windows services but does not allow you to specify the start type for a Windows service.

C: This class should not be used because it has a method that affects all services in an executable.

**QUESTION 49**

You work as the application developer at Certkiller .com. Certkiller .com uses Visual Studio.NET 2005 as its application development platform.

You are developing a .NET Framework 2.0 Windows Service application and are busy writing the following installation code for the Windows service:

```
<RunInstallerAttribute(True)> Public Class TestServiceInstaller Implements Installer
'Additional code to go here
End Class
```

You are required to install the Windows service and write the values associated with the service in the Windows Registry.

What should you do?

- A. The Assembly Registration Tool (Regasm.exe) should be used.
- B. The Global Assembly Cache Tool (Gacutil.exe) should be used.
- C. The .NET Services Installation Tool (Regsvcs.exe) should be used.
- D. The **Installer Tool (InstallUtil.exe)** should be used.

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

Explanation: The InstallUtil.exe utility is the correct toll that should be used because it will install the classes in the specified assembly that derive from the installer class and have the RunInstallArtrribute attribute set to True.

Incorrect Answers:

A: This tool should not be used in the scenario as this tool registers .Net assemblies so that COM clients can access .NET classes.

B: This tool should not be used as it is used to install and uninstall assemblies from the global assembly cache.

C: This tool should not be considered for usage as the tool is used for installing classes for Enterprise Services.

**QUESTION 50**

You work as the application developer at Certkiller .com. Certkiller .com uses Visual Studio.NET 2005 as its application development platform.

You are developing a .NET Framework 2.0 Windows service application that is used to perform several short tasks that require background processing.

You are not required to actively manage threads in your application but you are required to ensure that security checks are performed during the execution of the task.

What should you do?

- A. Use ThreadPool.**queueUserWorkItem**.
- B. Use ThreadPool.UnsafeQueueUserWorkItem.

- C. Use Thread.Resume.
- D. Use Thread.Start.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

Explanation: The scenario requires several short tasks to be run which requires background processing. Using the ThreadPool class is ideal in this situation as the system manages the thread pool meaning you have less overhead involved.

Incorrect Answers:

B: This method should not be considered for use as the security check the scenario requires will not be performed.

C, D: This method should not be considered because the Thread class is useful for creating threads that run in the foreground that are actively managed.

**QUESTION 51**

You work as the application developer at Certkiller .com. Certkiller .com uses Visual Studio.NET 2005 as its application development platform.

You are developing a .NET Framework 2.0 Windows application and need to create a foreground thread to draw lines in your application.

When the thread is started you must provide a data value that specifies the number of lines drawn.

What should you do? (Choose two)

- A. A WaitCallback delegate should be created
- B. A ThreadStart delegate should be created
- C. A ParameterizedThreadStart delegate should be created
- D. The ThreadStart method should be called
- E. The ThreadPool.QueueUserWorkItem method should be called

**Answer:** CD

**Section:** (none)

**Explanation/Reference:**

Explanation: The first order of operations would be to create a ParameterizedThreadStart delegate then use the ThreadStart method to start the thread. The ThreadStart method can be used to create actively managed foreground threads. The Parameterized ThreadStart delegate allows the passing of data when starting a thread.

Incorrect Answers:

A: This method should not be used as this method is for executing in a background thread that is part of a thread pool.

B: The ThreadStart delegate should not be used because you first need to pass parameters when starting a thread.

E: The method in this option should not be used because this method is used to create system managed background threads.

**QUESTION 52**

You are developing a .NET Framework 2.0 Windows Forms application on Certkiller -WS554.

The application must provide support for multiple languages and regional differences.

You are required to define a custom culture based on an existing culture and region.

An administrative user will install the custom culture on the end user's computer prior to the applications deployment

and you are required to select which class to use.

What should you do?

- A. The **CultureAndRegionInfoBuilder** class should be used
- B. The CustomAttributeBuilder class should be used
- C. The RegionInfo class should be used
- D. The CultureInfo class should be used

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

The correct option in the scenario would be to make use of the CultureAndRegionInfoBuilder class as this class is used to define a custom culture that is new or based upon an existing region and culture.

Incorrect Answers:

B: This class should not be used in the scenario because this class is used to define custom attributes which are used to associate declarative information.

C: This class should not be used in the scenario because this class is used to access the region data for an already installed culture.

D: This class should not be used in the scenario because this class can only be used to make use of cultures that have already been installed.

· CultureAndRegionInfoBuilder

**QUESTION 53**

You work as the application developer at Certkiller .com. Certkiller .com uses Visual Studio.NET 2005 as its application development platform.

You have recently finished development of a class named TestReward and package the class in a .NET 2.0 assembly named TestObj.dll.

After you ship the assembly and it is used by client applications, you decide to move the TestReward class from TestObj.dll assembly to the TestRewardObj.dll Assembly. You are to ensure when you ship the updated TestObj.dll and TestRewardObj.dll assemblies that the client applications continue to work and do not require recompiling. What should you do?

- A. The **TypeForwardedTo** attribute should be used
- B. The TypeConverter.ConvertTo method should be used
- C. The InternalsVisibleTo attribute should be used
- D. The Type Converter.ConvertFrom method should be used

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

Explanation: The statement used for you to add a type from one assembly into another assembly is the TypeForwardTo attribute which enables you not to have the application recompiled.

Incorrect Answers:

B, D: The TypeConverter class provides a unified way of converting different types of values to other types and can not be used to move a type.

C: The method in question here specifies all nonpublic types in an assembly are visible to other assemblies but can not be used to move types.

**QUESTION 54**

Certkiller .com has been hired by a small local private school to develop a class library that will be used in an

application named ManageAttendance for the purpose of managing student records.

You are responsible for developing this class library. Certkiller .com has instructed you to create a collection in the application to store learners' results.

The school has informed you that they currently only have seven learners, but that this value will triple in the following year.

Due to the limited resources, you need to ensure that the collection you create consumes a minimum amount of resources.

What should you use to create the collection?

- A. The HybridDictionary collection class.
- B. The HashTable collection class.
- C. The ListDictionary collection class.
- D. The StringCollection collection class.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

Explanation: You should use the HybridDictionary class to create the collection because this class is useful in scenarios where the number of elements is unknown or could grow in size. A collection of the HybridDictionary type manages the collection depending on the number of elements. The HybridDictionary type collection uses the ListDictionary class to manage the collection when there are only a few elements. When the number of elements exceeds ten, the HybridDictionary type collection automatically converts the elements into HashTable management.

Incorrect Answers:

B: You should not use this collection class because this class is used if the total number of elements to be stored in a collection is known and is greater than ten elements in length.

C: You should not use this collection class because this class is used if the total number of elements to be stored in a collection is known and is less than ten elements in length.

D: You should not use this collection class because this class is used to manage a collection of string values.

· Wie die Frage wenig aber später vielleicht mehr - Hybrid.

**QUESTION 55**

You are creating a class that uses unmanaged resources.

This class maintains references to managed resources on other objects.

You need to ensure that users of this class can explicitly release resources when the class instance ceases to be needed.

Which three actions should you perform?

(Each correct answer presents part of the solution. Choose three.)

- A. Define the class such that it inherits from the WeakReference class.
- B. Define the class such that it implements the IDisposable interface.
- C. Create a class destructor that calls methods on other objects to release the managed resources.
- D. Create a class destructor that releases the unmanaged resources.
- E. Create a Dispose method that calls System.GC.Collect to force garbage collection.
- F. Create a Dispose method that releases unmanaged resources and calls methods on other objects to release the managed resources.

**Answer:** BDF

**Section:** (none)

**Explanation/Reference:**

Explanation:

It is necessary to implement the `IDisposable` interface if you need to release unmanaged resources or want explicit control of the life of managed resources. A class destructor should be created to release the unmanaged resources and this should be called from within the `Dispose` method. The `dispose` method should also release the managed resources.

Inheriting from `WeakReference` would result in the garbage collector releasing resources even though there may be valid references.

The managed resources should be released in the `Dispose` method.

`System.GC.Collect` could be used, however it is more efficient to manually release the managed resources. The GC incurs overhead and may have only recently been called anyway. The question states resources should be released explicitly.

· D statt C - da im Destruktor die `Dispose`-Methode aufgerufen wird, und dadurch indirekt die unmanaged resources freigegeben werden.